

VARIABLES

VAR_GLOBAL

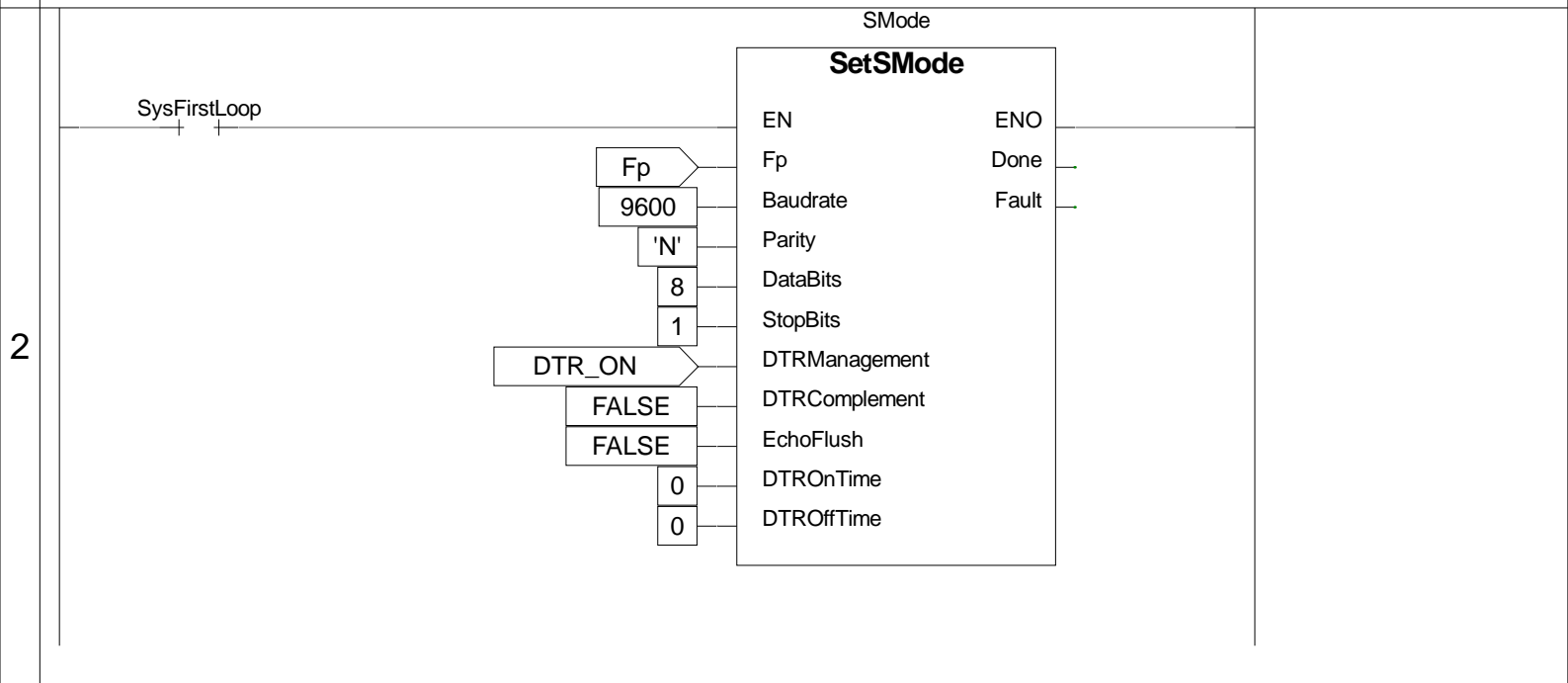
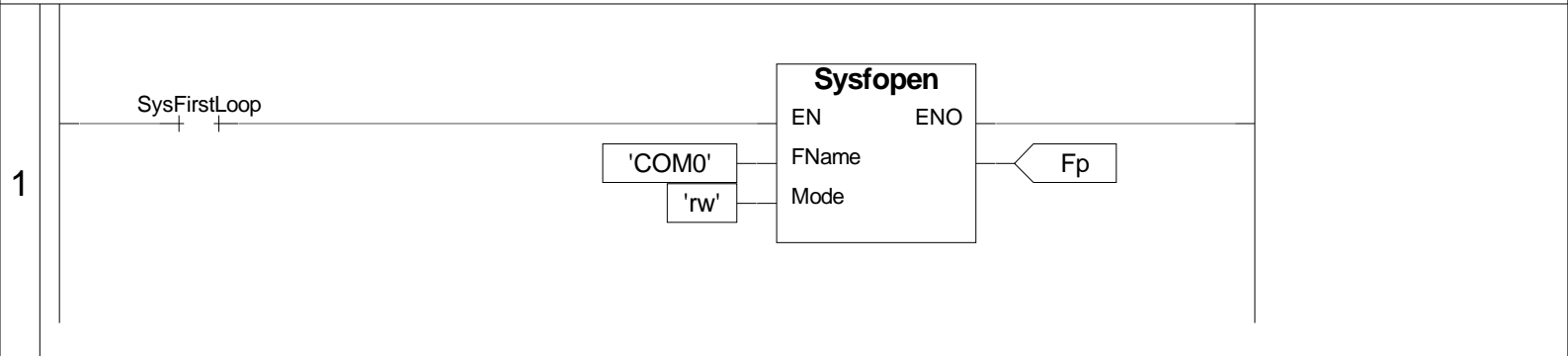
Di00M00 AT %IX0.0 : BOOL; (* Digital input 0 on module 0 *)

END_VAR

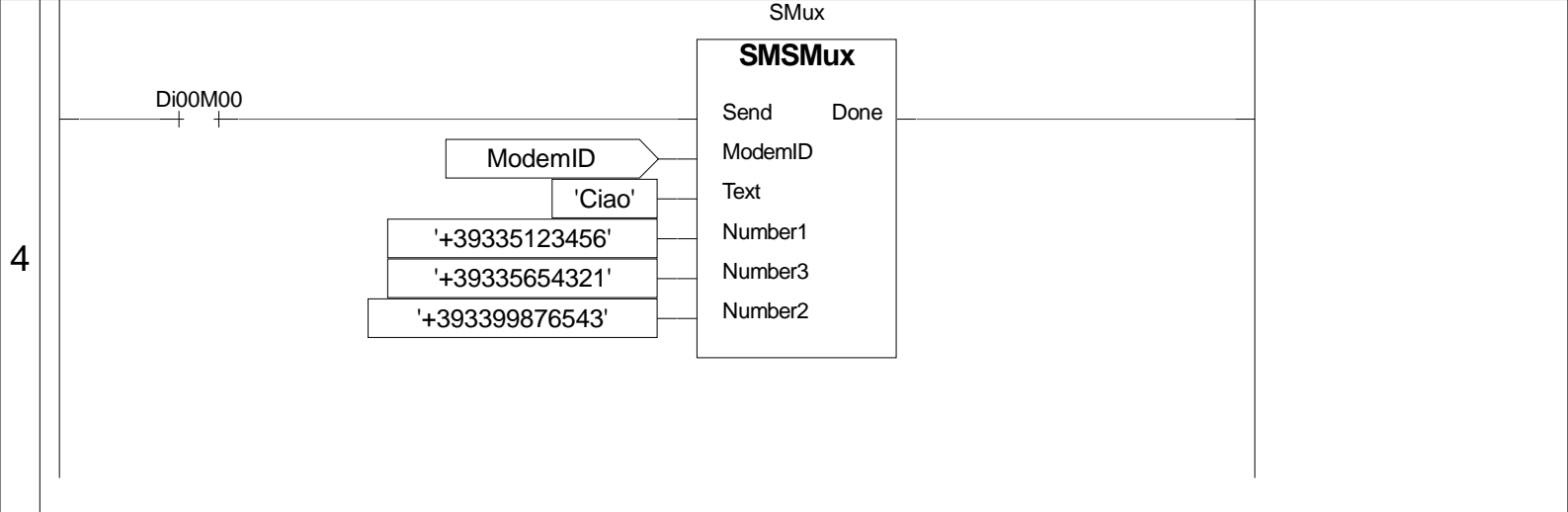
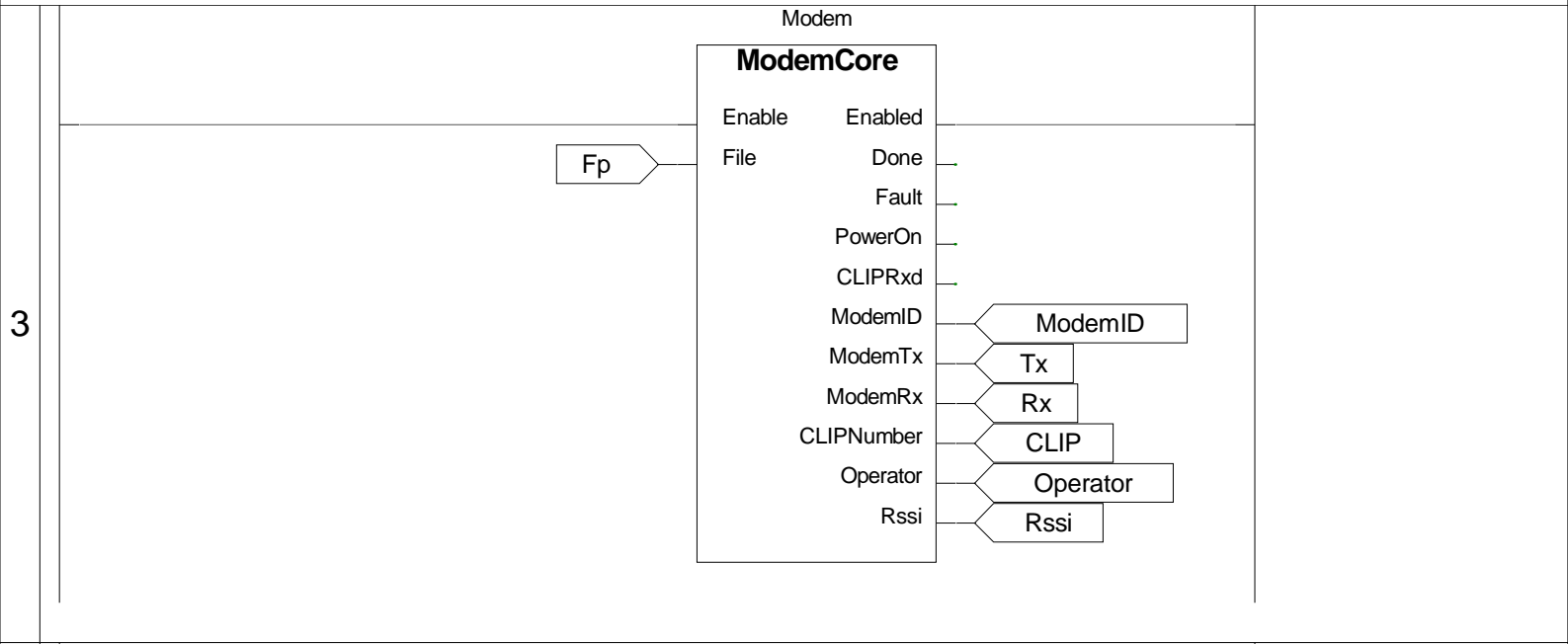
	Project : SMSMultiplexer	
	VARIABLES :	
	Release :	Ver :1.00
	Author :	Date:16/12/2011
	Note :	Page:1 of 1

```

VAR
CLIP : STRING[ 32 ]; (* CLIP received number *)
Fp : FILEP; (* Serial pointer *)
Modem : ModemCore; (* Modem core FB *)
ModemID : UDINT; (* Modem ID *)
Operator : STRING[ 16 ]; (* Operator value *)
Rssi : USINT; (* RSSI value *)
Rx : STRING[ 32 ]; (* Modem Rx string *)
SMode : SetSMode; (* Serial mode FB *)
SMux : SMSMux;
Tx : STRING[ 32 ]; (* Modem Tx string *)
END_VAR
    
```



Project : SMSMultiplexer	
PROGRAM : LDProgram	
Release :	Ver :1.00
Author :	Date:16/12/2011
Note :	Page:1 of 2



Project : SMSMultiplexer	
PROGRAM : LDProgram	
Release :	Ver :1.00
Author :	Date:16/12/2011
Note :	Page:2 of 2

Set the serial communication mode
ENCRYPTED CODE

```
VAR_INPUT
Fp : FILEP;
Baudrate : UDINT; (* Baudrate *)
Parity : STRING[ 1 ]; (* Parity type *)
DataBits : USINT; (* Nr of data bits *)
StopBits : USINT; (* Nr of stop bits *)
DTRManagement : USINT; (* DTR management type *)
DTRComplement : BOOL; (* Complement the DTR signal *)
EchoFlush : BOOL; (* Flush the echo *)
DTROnTime : UINT; (* DTR On wait time *)
DTROffTime : UINT; (* DTR Off wait time *)
END_VAR

VAR_OUTPUT
Done : BOOL; (* Execution done *)
Fault : BOOL; (* Execution fault *)
END_VAR
```

1

	Project : SMSMultiplexer	
	FUNCTION BLOCK : SetSMoDe	
	Release : SMSMultipl	Ver :1.00
	Author :	Date:16/12/2011
	Note :	Page:1 of 1

```

VAR_INPUT
Send : BOOL; (* Send command *)
ModemID : UDINT; (* Modem ID *)
Text : STRING[ 160 ]; (* Messaggio SMS da inviare *)
Number1 : STRING[ 16 ]; (* Numero telefonico (1°) *)
Number2 : STRING[ 16 ]; (* Numero telefonico (2°) *)
Number3 : STRING[ 16 ]; (* Numero telefonico (3°) *)
END_VAR

```

```

VAR_OUTPUT
Done : BOOL; (* Invio eseguito *)
END_VAR

```

```

VAR
PhNr : USINT; (* Sequencer numeri di telefono *)
SdCmd : BOOL; (* Comando invio SMS *)
SMSSend : ModemSMSSend; (* FB invio SMS *)
StartPls : BOOL; (* Start pulse *)
END_VAR

```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "SMSMux" *)
3 (* ***** *)
4 (* Il blocco funzione esegue l'impostazione dei parametri di comunicazione *)
5 (* su porta seriale. *)
6 (* ----- *)
7 (* Eseguo inizializzazioni. *)
8
9 Done:=FALSE; (* Invio eseguito *)
10 SMSSend(); (* Eseguo FB invio SMS *)
11
12 (* Memorizzazione comando invio SMS, il comando deve essere impulsivo. *)
13
14 IF (Send) THEN StartPls:=TRUE; SdCmd:=TRUE; END_IF;
15 IF NOT(StartPls) THEN PhNr:=0; RETURN; END_IF;
16
17 (* ----- *)
18 (* SELEZIONE NUMERO DI TELEFONO *)
19 (* ----- *)
20 (* Eseguo selezione numero di telefono. *)
21
22 CASE PhNr OF
23 0: SMSSend.Number:=Number1; (* Phone number *)
24 1: SMSSend.Number:=Number2; (* Phone number *)
25 2: SMSSend.Number:=Number3; (* Phone number *)
26 ELSE
27 Done:=TRUE; (* Invio eseguito *)
28 StartPls:=FALSE; (* Start pulse *)
29 RETURN;
30 END_CASE;
31
32 (* ----- *)
33 (* GESTIONE INVIO SMS *)
34 (* ----- *)
35 (* Eseguo invio SMS. *)
36
37 SMSSend.ModemID:=ModemID; (* Modem ID *)

```

Project : SMSMultiplexer	
FUNCTION BLOCK : SMSMux	
Release : SMSMultipl	Ver :1.00
Author :	Date:16/12/2011
Note :	Page:1 of 2

FUNCTION_BLOCK SMSMux

```

38  SMSSend.Text:=Text; (* Testo messaggio *)
39  SMSSend.Send:=SdCmd; (* Comando invio SMS *)
40  SdCmd:=FALSE; (* Comando invio SMS *)
41
42  (* Eseguo attesa fine invio. *)
43
44  IF ((SMSSend.Done) OR (SMSSend.Fault)) THEN
45      SdCmd:=TRUE; (* Comando invio SMS *)
46      PhNr:=PhNr+1; (* Sequencer numeri di telefono *)
47  END_IF;
48
49  (* [End of file] *)
50
51

```

	Project : SMSMultiplexer	
	FUNCTION BLOCK : SMSMux	
	Release : SMSMultipl	Ver :1.00
	Author :	Date:16/12/2011
	Note :	Page:2 of 2